UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPRATUS OF LOWERING I/O BUS POWER CONSUMPTION

INVENTORS:

VICTOR W. LEE

PHANINDRA K. MANNAVA

AKHILESH KUMAR

SANJAY DABRAL

DOCKET NO. 42P17402

PREPARED BY:

AMI PATEL SHAH

REG. NO. 42,143

A METHOD AND APPARATUS OF LOWERING I/O BUS POWER CONSUMPTION

BACKGROUND INFORMATION

[0001] Many mechanisms have been developed to manage electronic device/component power. Intel and other companies have drafted an ACPI (Advanced Configuration and Power Interface) specification which uses multistage approach to scale power consumption with usage. However, the ACPI specification does not provide the actual implementation for power management. This leads to individual hardware providers design to implement their own power management methods.

[0002] In the area of interface power management, two common methods are: 1) lowering the I/O interface frequency and 2) turning off the entire I/O interface when not used. Changing the interface frequency on the fly is complicated and a large setting time is required to stabilize the interface after the frequency change. Turning on the interface from the power off mode requires a complete re-initialization of the entire interface. Therefore, a need exists for a method to maintain basic interface functionalities such as passing credits/acks and provide CRC checksum to link control bits in power saving mode.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Various features of the invention will be apparent from the following description of preferred embodiments as illustrated in the accompanying

drawings, in which like reference numerals generally refer to the same parts throughout the drawings. The drawings are not necessarily to scale, the emphasis instead being placed upon illustrating the principles of the inventions.

[0004] Figure 1 is a block diagram of a transmitter sending a low power command to a receiver.

[0005] Figure 2 is a flowchart representing the operation of the transmitter and receiver transitioning into low power mode.

[0006] Figure 3 is a flowchart representing the operation of the transmitter and receiver waking up from low power mode.

[0007] Figure 4 is an example of a packet in low power mode.

[0008] Figure 5 is an example of the packet in Fig. 4 waking up from low power mode.

DETAILED DESCRIPTION

[0009] In the following description, for purposes of explanation and not limitation, specific details are set forth such as particular structures, architectures, interfaces, techniques, etc. in order to provide a thorough understanding of the various aspects of the invention. However, it will be apparent to those skilled in the art having the benefit of the present disclosure that the various aspects of the invention may be practiced in other examples that depart from these specific details. In certain instances, descriptions of well-known devices, circuits, and methods are omitted so as not to obscure

the description of the present invention with unnecessary detail.

[0010] This application is regarding I/O buses that connect different components together in a computer system. The type of I/O buses the present application is concerned with are known as links. A link is a point-to-point interconnect connecting two components (these components can be on the same circuit board or across two different boards). A link is always bi-directional and consists of an out-going direction and an in-coming direction. The width of the link is scalable from one bit (a.k.a. serial) to multiple bits in parallel. A single bit may be transferred from the source component via a transmitter and received at the destination via a receiver. In multi-bit parallel links, multiple bits are transferred simultaneously in parallel through multiple transmitter and receiver pairs. This signaling technology can be single ended or differential.

[0011] The power consumed by a link scales almost linearly with the width of the link (i.e. the number of serial I/O channels). The power also scales with the frequency of the I/O channels. Thus, a significant portion of the I/O channel is consumed by the transmitter and receiver pair. For example, a 16 bit bi-directional I/O bus running at 3.2 GT/s can easily consume 2 W of power. When multiple of such I/O buses are integrated into a component, the I/O power consumption can take up a significant portion of the components' power budget. As an example, for a CPU with 6 links, the power budget for I/O buses could be 12 W or 10% of a 120 W CPU thermal budget. This does not include the power for the Link and Protocol stack. By having coordinated

shut down of certain link components can easily save 1 W of power per link.

[0012] As previously stated, each link consumes a significant amount of power. Turning the links on and off is not like a switch that can turn on or off automatically. They are protocols the link has to follow before turning on or off such as, for example, conserving the current state. Since links are analog signals, there is a lengthy initialization sequence that must be followed to turn a link on if it is in the off state. Link initialization may include: electrical calibration, clock synchronization, channel to channel diskewing, framing, and synchronization of operating parameters. This initialization sequence can take up to millions of cycles to complete. By contrast, the current protocol allows the link to power up in tens of cycles.

[0013] Packets of communication used in high speed interconnects consists of a command portion and a data portion. When a packet is idle, the data portion is not used by the upper layers (protocol/system). Therefore, the current invention has the ability to optionally switch off those transmitter and receiver pairs associated with the data portion that is turned off. Power saving is achieved by selectively turning off these non-essential parts in a very low latency wake up mode. The benefits include allowing power scaling for I/O bus based on utilization, improved component power management and in-band power management signaling.

[0014] Referring to Fig. 1, a transmitter 10 includes a link activity monitor 15. The activity monitor 15 monitor's the activity on this particular link. The monitor 15 notifies the control logic on the link to take some action based on

what the monitor 15 has detected. If there is no activity on this link for a period of time the control logic sends a signal to a receiver 20. This signal command is known as a sleep command 25. The receiver 20 acknowledges receiving the sleep command 25 to the transmitter 10 and then both the transmitter 10 and the receiver 20 go into sleep mode. It should be noted that power management can be independently applied to either direction of the link. Thus, in the present invention, power management can be applied to each direction of traffic independently. As an example, the present application has the sleep command being sent by the transmitter to the receiver.

[0015] The sleep command 25 is sent by the link. The command 25 can be, for example, a packet made of 80 bits, 20 links, 4 transfers per cycle. The size of the link can be changed to fit the implementation. When the transmitter 10 sends the sleep command 25 to the receiver 20, the receiver 20 determines if anything is still transmitting. In the meantime, the transmitter 10 stops transmitting and the receiver 20 waits to send acknowledgment until it has saved all its buffers. Once all the buffers are saved, the receiver 20 goes to sleep and sends an acknowledgement to the transmitter 10 that it is going to sleep. Once the transmitter 10 receives the acknowledgement from the receiver 20, the transmitter 10 then goes into low power mode.

[0016] Referring now to Figs. 2 and 4, where Fig.2 is a flowchart illustrating one sequence of operation of the transmitter and receiver pair going into low power mode and Fig. 4 is the corresponding packet when in low power mode. To maintain high error detection capabilities, CRC is still

computed over the whole message, 80 bits, while assuming the bits in the link that are tuned off to have a value of zero. As shown in Fig. 4, 16 of the wires may be turn off and 4 of the wires may be still on. It should be noted that there are multiple ways of implementing the link and that CRC is just one implementation and other implementations may be used, such as ECC.

[0017] Beginning with step 200, the data link layer on the transmitter 10 may receive a signal from the link activity monitor 15 or the upper layers of the transmitter10 to place the link in light sleep mode. The transmitter then sets the bits in the unused portion of the packet to zero and computes the CRC checksum before transmitting the packet. Once the transmitter 10 receives the command 25, the transmitter 10 sends the sleep command 25 to the receiver 20 (step 210). When the receiver 20 receives the sleep command 25, the receiver 20 will makes some assumptions. First, the receiver 20 will assume that the input is 0 on the 16 wires. In step 220, the receiver 20 computes a CRC checksum calculation to see if there are any errors on the link. The receiver needs to check for errors because the link is never idle, there is always something being sent on the link.

[0018] CRC is computed in a message basis (both on the transmitter and receiver ends). The transmitter computes the CRC and transmits it as part of the command portion and the receiver recomputes the CRC and compares it with the transmitted CRC to see if any transmission error occurs (step 230). In step 240, the receiver 20 uses well know error detection methods. In particular, the receiver 20 assumes input flit payload to be zeros and

continues. The receiver 20 then goes into light sleep mode and now turns off the power on the receiver 20 (step 250). Once the receiver 20 is in low power mode, the receiver can send an acknowledgement signal to the transmitter 10 that it is now in low power mode (step 260). Otherwise, if the transmitter 10 has not received an acknowledgement signal from the receiver 20 after some period of time, the transmitter 10 will go into sleep mode by itself (step 270).

[0019] Referring now to Fig. 3, a flowchart illustrates one sequence of operation of the transmitter and receiver pair waking up from low power mode. Beginning with step 300, the data link layer of the transmitter 10 may receive a wake up command from either the link activity monitor 15 or the upper layers of the transmitter 10. Upon receiving the wake up command, the transmitter 10 wakes up (using the current example) all 16 wires in step 310. There are only 16 wires to turn on because 4 of the wires were never turned off as shown in Fig. 4. At this time, all data payloads are still assumed to be zero.

[0020] Once all 16 wires are on, the transmitter 10 will change the pattern in the 4 wires that were left on (step 320). As shown in Fig. 4, the 4 wires that were left on has a particular pattern. In this instance this pattern was 00,11,00,11. It should be noted that the wires can be any pattern for the implementation. Now that the transmitter 10 is waking up from sleep mode, the transmitter 10 will change the pattern of the 4 wires that were left on. In this instance, as shown in Fig. 5, the transmitter 10 has changed the pattern to 00,00,11,11. Once the receiver 20 receives the new pattern, the receiver 20 knows by comparing the original pattern to the new pattern that the transmitter

10 is awake and wakes up the receiver 20 (step 330). In step 340, once the receiver 20 wakes up, the receiver 20 can send an acknowledgement signal to the transmitter 10 or after a period of time, the transmitter 10 will start transmitting its data.

[0021] In the method disclosed, the interface actually behaves like it is in the idle mode. Therefore, from the upper communication layers (protocol layer, system firmware, system OS) perspective, the link is still active. This reduces software complexity. Moreover, keeping the link alive during power saving mode allows the link to maintain its operation (such as passing credits/acks back and forth between agents as well as providing CRC checksum against transmission error). These features are unique to the method disclosed above and do not exits in current methods.

[0022] The present invention provides a novel approach to manage the power consumption of a high speed I/O interface by selectively turning off non-essential portion of the interface. Here only part of the interface is powered off as compared to the whole interface being turned off and by keeping part of the interface on, the current method maintains the interface operation state. Thus, from the upper layers (protocol/system) perspective, the interface is always "on".

[0023] The current method further allows the link to scale back for just enough bandwidth to maintain the link during idle (by turning off the non-essential parts). Only a small portion of the link bandwidth is required to maintain the link during idle (only credits and acks are needed to pass back

and forth). Thus the present method provides for greater power efficiency.

[0024] Since the link is still operating in full speed (only in a scaled back fashion) the link may return to full bandwidth operation in a matter of ten cycles. Furthermore, the link wake up latency can be completely hidden from the upper link layers. This may be accomplished by programming the data link to wake up the physical layer as soon as it receives a request from the protocol layer. This way, the physical layer can perform link wake up protocol while the data link layer process the request.

[0025] Advantageously, the above method provides significant power savings, sometimes up to greater than 80%. The method is applicable to any high speed I/O interface. Furthermore, the power saving mode described herein is particularly suitable for inclusion in mobile technology providing significant power saving and low wake-up latency.

[0026] In the following description, for purposes of explanation and not limitation, specific details are set forth such as particular structures, architectures, interfaces, techniques, etc. in order to provide a thorough understanding of the various aspects of the invention. However, it will be apparent to those skilled in the art having the benefit of the present disclosure that the various aspects of the invention may be practiced in other examples that depart from these specific details. In certain instances, descriptions of well-known devices, circuits, and methods are omitted so as not to obscure the description of the present invention with unnecessary detail.